

REMARKS/ARGUMENTS

In the Office Action mailed on March 10, 2006, the Examiner rejected Claims 1, 3-9, 12, 14-16, 19-20, 24, 26, 28, 30-33 and 35-51 under 35 U.S.C. §102(e) as anticipated by U.S. Patent No. 6,205,575 to Sherman ('575). Examiner also rejected claims 10, 17, 22, 29 and 34 as unpatentable over Sherman ('575) in view of Werner et al., and claims 11, 13, 23, 25 and 27 as unpatentable over ('575) in view of Ladkin et al.

The present response is intended to be fully responsive to all points of objection and/or rejection raised by the Examiner and is believed to place the application in condition for allowance. Favorable reconsideration and allowance of the application is respectfully requested.

Applicants assert that the present invention is new, non-obvious and useful.

Status of Claims and Support for Changes in the Claim Listing

Claims **1, 3-17, 19-20 and 22-51** were pending.

Claims **1, 3-17, 19-20, and 22-51** were rejected.

Claim **9** has been cancelled by this Amendment.

Claims **1, 3-6, 8, 14-16, 19-20, 38-39** are currently amended.

Claim **52** is newly added.

Claims **1, 3-8, 10-17, 19-20, 22-52** are currently pending in the application

Any amendments and cancellations detailed here should not be construed to have been made in order to overcome prior art unless specified otherwise.

Claims have been amended or added in order to clarify the subject matter of the application. In making these revisions and additions, care has been taken to ensure that no new matter has been added. The lines from the original specification are stated and/or reproduced below solely to demonstrate that no new matter has been added, and therefore no limitations should be read into the claims based on these stated and/or reproduced specification lines.

The claim amendments and new claims are supported *inter-alia* by the originally filed specification, as follows:

Independent claims 1 and 19 have been amended to recite that a graphic user interface (GUI) which is provided for play-in includes at least one object with at least one property which does not change in reaction to an input unless change is specified during playing in. This amendment is supported in the specification, *inter-alia* on page 7, lines 29-31:

Before the user can start with the play-in procedure, he/she is provided (either by building in on his/her own or receiving it from another source) with a GUI of the system (**21** in Fig. 2A) that has no logic built into it.

Independent claims 1 and 19 have also been amended to recite that description(s) of input(s) and description(s) of change(s) in property (properties) of object(s) in reaction to at least some of the inputs received during playing-in. This amendment is supported by the specification, *inter-alia*, on page 8, line 26 to page 9, line 6:

A typical, yet not exclusive, scenario of using the specified object operation for system behavior specification would be that the user specifies actions (e.g. operating objects such as clicking buttons), and "describes" the system reaction and optionally the assignments/condition(s) that may or must hold regarding one or more of the operated objects, using the same techniques (e.g. operating objects – such as displaying the so clicked value on the display and storing the so displayed value for future use). The objects operated in the system reaction mode may be the same, different or partially overlapping to those of the user action mode, depending upon the particular application. This procedure may be repeated as many times as required to accomplish the entire playing-in scenario.

Independent claims 20 and 39 have been amended to recite that a graphic user interface (GUI) which is provided for play-out includes at least one object with at least one property which does not change in reaction to an input except in accordance with a formal behavior specification. This amendment is supported by the specification, *inter-alia* on page 13, lines 28-34:

In accordance with an aspect of the invention, options are provided for playing-out a scenario. After specifying a part of the system behavior, or all of it, the user may start working with the GUI application as if it was an operational application and, if desired, it can actually function and the final system. The user will operate the objects as described before and the Play-Engine will follow the user actions and will make the application react to these actions according to the previously defined specification.

Independent claims 20 and 39 and dependent claim 14 have also been amended to recite that during playing out a description of at least one input is received. This amendment is supported *inter-alia* on page 47, lines 5-12:

Bearing in mind the description of the functions above, the procedure, in accordance with this embodiment, for playing out a scenario (2904) by the user is relatively simple. After the user enters play-out mode, he/she operates the system by demonstrating user and external environment actions through the system GUI. Each such action is considered by the play-engine as a step. After this step is executed, the procedure performs a following super step. This super step consists of all the system reactions that should follow the user action, according to the participating universal charts.

Independent claims 1 and 19 have been amended to recite that a property of an object changes in accordance with the received description of change. This amendment is supported by the specification, *inter-alia*, on page 9, lines 12-16, and on page 28, line 34 to page 29 line 1:

Preferably, the application's state during the play-in operation is reflected in the GUI application. For example, consider an output object that is being operated (say, a light indicator that is turned on) the result is reflected in the system GUI. (i.e., the light indicator is portrayed in its "turn on" state).

The play-engine also sends a message to the GUI application, telling it to change the object's property in the GUI itself so that it reflects the correct value after the actions were taken

Independent claims 20 and 39, and dependent claim 14 have been amended to recite that when playing-out, property/ies of object(s) change in reaction to at least some inputs in accordance with a formal system behavior specification. This amendment is supported *inter-alia* on page 13, lines 28-34:

In accordance with an aspect of the invention, options are provided for playing-out a scenario. After specifying a part of the system behavior, or all of it, the user may start working with the GUI application as if it was an operational application and, if desired, it can actually function and the final system. The user will operate the objects as described before and the Play-Engine will follow the user actions and will make the application react to these actions according to the previously defined specification.

Claims 5 and 38 have been amended to recite that the description may be provided through operating objects. The amendment is supported by the specification *inter-alia* on page 8, line 26 to 34:

A typical, yet not exclusive, scenario of using the specified object operation for system behavior specification would be that the user specifies actions (e.g. operating objects such as clicking buttons), and "describes" the system reaction and optionally the assignments/condition(s) that may or must hold regarding one or more of the operated objects, using the same techniques (e.g. operating objects – such as displaying the so clicked value on the display and storing the so displayed value for future use). The objects operated in the system reaction mode may be the same, different or partially overlapping to those of the user action mode, depending upon the particular application.

Other claim amendments are of a more formal nature, for example, providing proper antecedent basis.

New claim 52 recites that more than one chart may specify an object property change in reaction to the same input which is taken into account during playing-out. Support for this amendment is found *inter-alia* on page 37, lines 30-34:

Typically, although not necessarily, an LSC specification includes of several LSCs [3]. A single universal chart may become activated (i.e., its prechart is successfully completed) several times during a system run. Some of these activation's might overlap, resulting in a situation where there are several "live copies" of the same chart active simultaneously.

and on page 39 lines 30-34:

Non-determinism arises from the interleaving characteristics of the execution algorithms. While executing LSCs, there may be several cases where the choice of what should be done next is non-deterministic. One example of such a choice is simply choosing the next event to be activated from a set of some possible charts.

and on page 40, lines 11-21:

This non-determinism concerns the order in which events are searched through the specification. The user may define an *LSC search list* and a search policy. The search list dictates the order in which LSCs are searched for the next event to be activated. The policy is a simple flag that may be assigned a value from a so called *Deep*, *Wide* states. If the *Deep* policy is taken, the algorithm will always try to pick the next event from the first LSC in the list. If no such event may be taken, it will look in the second LSC and so on. In this way, the algorithm tries to complete the LSCs in the order specified by the user. If, on the other hand, the *Wide* policy is taken, the algorithm will try to pick one event from each LSC in the order they are located in the search list.

and on page 41, lines 19- 24:

The maximal approach tries to activate as many charts as possible and keeps them active as long as possible by avoiding the triggering of violating events. This approach is called maximal since it usually causes maximal reactions of the application. The opposite approach is the safe one. According to this approach, an attempt is made to violate as many precharts as possible, by choosing events from other charts that violate these precharts.

35 U.S.C. § 102 Rejections

The Examiner rejected claims 1, 3-9, 12, 14-16, 19-20, 24, 26, 28, 30-33, and 35-51 under 35 USC 102(e) as being anticipated by Sherman (US 6,205,575). Applicants appreciate the time and consideration provided by the Examiner in reviewing this application, however, respectfully traverses the rejection of the claims at least for the following reasons.

Anticipation under 35 U.S.C. 102 requires that each and every claimed feature be disclosed by a single prior art reference.

In order to assist the Examiner in better understanding the subject matter of the invention, Applicants submit herewith a demonstrational CD showing how the changes in the object properties occur in reaction to the inputs received during the Play-In.

The demonstrational CD shows one possible implementation of the invention and therefore any content including additions, limitations, etc. of the demonstrational CD which are not specified in the claims should not be read into the claims.

The demonstrational CD includes two executable files, one demonstrating play-in ("PlayInBasics") and one demonstrating play-out ("PlayOutCalc"). By clicking on PlayInBasics.exe, a demonstration of play-in begins. First, the setup for play-in according to one implementation is demonstrated, including opening the GUI application, adding a use case, adding a live sequence chart LSC (in this case corresponding to the turning on a calculator), and selecting a play-in mode. Next, the construction of the LSC is demonstrated. A description of an input is received when the On-Off button is pressed on. Description of any changes in properties of objects in reaction to the input are then received, including the change in the state of the light from Off to On, change in the state of the display from Off to On, and change in the background of the display to green. These described

changes also occur, with the light turning on, the display turning on, and the background of the display turning to green. On the left side of the screen, the construction of the LSC is demonstrated. Play-in is then stopped, and a natural language translation of the LSC is displayed. The demonstration then ends.

By double clicking on PlayOutCalc.exe, a demonstration of play-out begins. It is better to have a display resolution of at least 1152 x 864 to be able to see all that is happening on the display. First, the selection of the play-out mode is demonstrated. Next is demonstrated the play-out using both step mode and superstep mode (see page 34 lines 16-22 of the current application for a description of the difference between the step and the superstep modes). First the selection of step mode is demonstrated. A description of an input is received when the On-Off button is pressed on. In reaction to the input, the state of the light is turned on, the state of the display is turned on and the background of the display is turned to green, where the changes in these properties are in accordance with the "switch" LSC. Because of the step mode, the user is prompted after each event. Note that more than one LSC may be opened based on an input, for example, in this case both the "switch" LSC and the "InitSlider" LSC are opened. In this implementation, there is a slider which can be seen on the bottom of the calculator and which at initialization would have been set to zero if the slider had not already been at zero. The demonstration then switches to the superstep mode and a description of an input is received when the number 4 is pressed. In response, the value of the display is changed to 4, in accordance with the "ShowNumber" LSC. A description of another input is received when the multiplication symbol "x" is pressed. In response, the value of the slider is set to 10, in accordance with the "Mul-New" LSC. A description of another input is received when the 5 is pressed. In response, the value of the display is changed to 5, in accordance with the "ShowNumber" LSC. In addition, the value of the slider is reset to zero.

The step mode is then switched to On. A description of another input is received when the equal sign "=" is pressed. A reaction then occurs in accordance with the "MulBySum" LSC. The background of the display is changed to red. The value of the display is changed to 0. The value of the display is then changed sequentially to the sums 5, 10, 15, and 20. Once the final sum of 20 is reached (i.e. the product of 4 x 5), the background of the display is changed to green.

The superstep mode is then switched to On, for showing a multiplication of the result (i.e. 20 x 8. A description of an input is received when the multiplication symbol "x" is pressed. In response, the value of the slider is set to 10, in accordance with the "Mul-New" LSC. A description of another input is received when the 8 is pressed. In response, the value of the display is changed to 8, in accordance with the "ShowNumb" LSCs. In addition, the value of the slider is reset to zero. A description of another input is received when the equal sign "=" is pressed. In reaction , the value of the display changes to 160, in accordance with the "MulBySum" LSC. The demonstration ends when the play-out stops.

Based on the specification and claims of the application and having in mind the demonstration presented on the CD, Applicant respectfully traverses the rejection for at least for the following reasons.

One of the features recited in each of independent claims 1, 19, 20, and 39 of the current invention is that in reaction to a certain input, a property of an object of a graphic user interface (GUI) changes. During playing-in (claims 1 and 19) the change is as per a received description of the change. During playing-out (claims 20 and 39), the change is as per a formal behavior specification. Both for playing-in and playing-out, except for the received description of the change or formal behavior specification, the property of the object would not change in reaction to the input, because the GUI initially has no logic built into it. In many cases, the change in the object property can be sensed (e.g. viewed, heard, etc) by a user thereby allowing the user to get a better grasp of the reaction to the input. (See Demo CD).

In contrast, the animation in Sherman provides a pleasing visualization of the steps of the scenario definition process but the animation does not model the change in object properties in reaction to inputs. For example, in Column 6 lines 61-64 of Sherman, it is stated that the steps (of the defined scenario sequence) are animated:

In accordance with another aspect of the invention, a scenario is at least in part defined, including a step of sequential animation wherein the steps are animated sequentially by way of a graphical technique.

Again, from column 13, lines 50 to 55 of Sherman, it is clear that what is being animated in Sherman is the *steps of the sequence*:

At any point, the scenario can be animated. This means that the steps of the sequence are sequentially illustrated with highlighting and moving icons. The animation of a step is depicted in FIG. 5. It is noted that the step number has been highlighted and the value being sent is indicated in brackets below the name of the dataflow.

More details on the animation are provided in Sherman in Column 15, lines 44-59:

Animation refers to the graphical presentation in sequence of the participating elements of the diagram. Before any animation is started, it is necessary to check that the thing to be animated is visible on the screen. Because diagrams can be very large and they can be off screen, there can be practical problems of display. Accordingly, scrolling is first carried out through that portion of the screen of diagrams into the view so that they can be animated. Then the source node is animated, for example, either by blinking or writing in some fashion. Then the flow is animated in some graphical way, for example, either by moving an icon along the flow or by graphically changing the color of the flow showing some movement. Clearly there exist many choices in implementing animation by using a variety of graphical techniques. Thereafter, the destination is animated in similar fashion.

The above quote shows clearly that the animation used in Sherman does not even try to attempt to model what is happening to objects (i.e. changes in properties). The animation techniques function

solely as "special effects" for displaying the steps of the scenario definition process and therefore there is no limitation to how the animation is affected. That is why the above quote states "*clearly there exist many choices in implementing animation by using a variety of graphical techniques*"

In contrast, in the present invention, the change in the object property is recited as being in accordance with the received description of the change or in accordance with the formal behavior specification, so as to provide a faithful model of the system reaction.

Again, from column 3, lines 22-28 of Sherman, it is clear that the animation of Sherman is no more than a graphical technique for illustrating the step by step nature of the scenario:

Animation is provided to illustrate the step-by-step nature of a scenario. The nodes and flows provide graphical techniques for depicting emitting a message, receiving a message, and transferring a message. Use of animation leads the user through the scenario, and eliminates translating between a textual description and lifeless static diagrams.

Nowhere in the above quotes from Sherman nor elsewhere in Sherman is there a hint or an indication that Sherman's animation is capable of modeling the change in object properties in reaction to inputs. Moreover, in Column 6, lines 5 to 30, Sherman explicitly states that the animation described by the patent does not include system execution:

Animation vs. Simulation

Animation is visually appealing. It is always interesting to see something come to life as a result of a design effort. Visual verification helps form intuition and understanding which can be applied in all steps of the design process. In considering the question of what can be animated, it is noted that animation requires at least a set of entities, a set of transitions or dataflows, and some control. In the early phases the control is often lacking, but this can be supplied by assertion. That is, the sequence of events can be postulated by the designer in a scenario that depicts what the system is supposed to do. This sequence of activity can then be captured and animated on the diagrams created to model the system.

More elaborate animations can show the allocation and deallocation of memory, variable updates, heap size, stack contents, file i/o, and other activities of interest. In object-oriented (OO) systems, object creation and deletion, and thread of control are of considerable interest. *However, to provide animation requires the ability to perform an actual simulation of the system. A simulation requires a significant amount of the system design before anything can be done at all. That is, it is required that the actual system control be defined in sufficient detail so that the system can be effectively "executed" in the simulation environment. Animation, in contrast, can be performed at much earlier stages of design.* (emphasis added)

Sherman clearly differentiates with the words "in contrast" between the animation, which Sherman performs and which Sherman says can be performed at early stages of design, versus simulation which demonstrates "system execution" and which Sherman says can only be done after a significant amount of system design. Therefore, not only does Sherman not anticipate the current invention but Sherman in effect teaches away from the current invention by stating that system execution can not be simulated in the early design stages, whereas in the current invention *system execution* is demonstrated in the *early design stage*, with object properties changing in reaction to inputs.

The demonstrational CD provided herewith shows that, in contrast to Sherman, where simulation of system execution in the early design stage *is not considered possible*, in the current invention changes in the object properties in reaction to the inputs in fact occur.

In view of the above, Applicants respectfully submit that the independent claims 1, 19, 20 and 39 and all claims depending directly or indirectly upon these claims are novel in view of the prior art.

Rejection under 35 U.S.C. §103(a)

Claims 10, 17, 22, 29 and 34 rejected under 35 U.S.C. §103(a) as unpatentable over Sherman in view of Werner et al., and claims 11, 13, 23, 25 and 27 as unpatentable over Sherman in view of Ladkin et al.

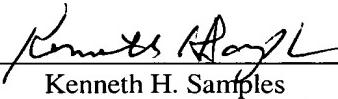
Since all rejections under 35 U.S.C. §103(a) are based primarily on Sherman, the arguments presented above are also applicable to the §103 rejection. As stated in MPEP 2142, in order to establish a *prima facie* case of obviousness, the references alone or in combination must teach or suggest all the claimed limitations. Applicants maintain that dependent Claims 10, 11, 13, 17, 22, 23, 25, 27, 29 and 34 should be deemed novel and non obvious over the cited references, *inter alia*, for the reasons discussed with reference to their respective independent base claims.

In view of the above, Applicants respectfully submit that all pending claims are allowable, and the application is now in condition for allowance, which allowance is earnestly solicited.

The Commissioner is hereby authorized to charge any fees, which may be required in connection with this correspondence, to Deposit Account No. 06-1135.

Respectfully submitted,

FITCH, EVEN, TABIN & FLANNERY

By 
Kenneth H. Samples
Registration No. 25,747

Date: July 10, 2006

120 South LaSalle Street
Suite 1600
Chicago, IL 60603-3406
Telephone: (312) 577-7000
Facsimile: (312) 577-7007